

Physics Production at the Fermilab LQCD Facility

J. Simone

CD Status Report

28 Nov 2006



Fermilab LQCD Facility is a Community Resource



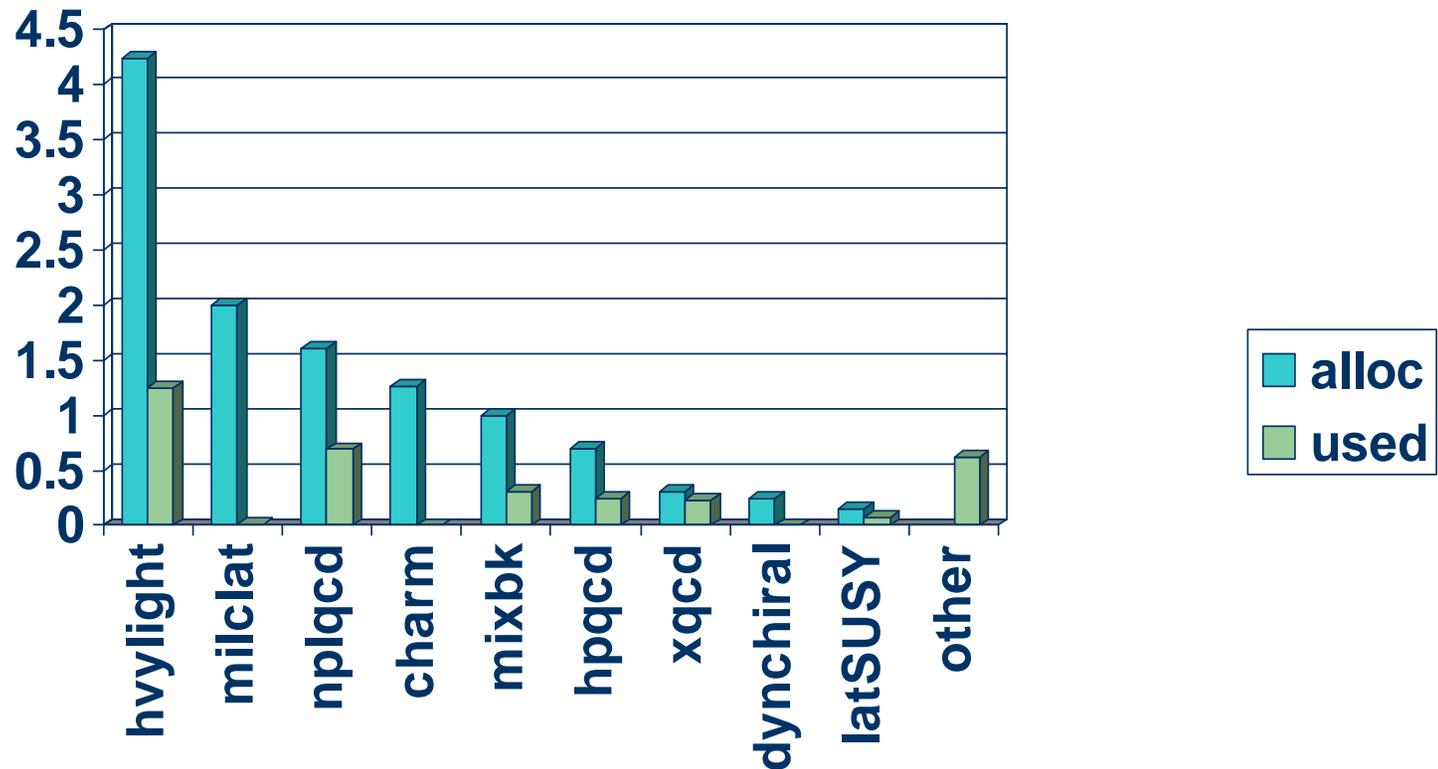
Resource allocations based on project proposals submitted to the USQCD Scientific Allocations Committee.

Ten allocated projects in 2006

- **Heavylight:** Mackenzie + 20 collaborators
- **Charmonium:** Kronfeld + 6 collaborators
- **MILC lattice:** Sugar + 10 collaborators
- **Nuclear Physics LQCD:** Savage + 3 collaborators
- **Mix B_K :** Laiho + 2 collaborators
- **HPLQCD:** Lepage + 13 collaborators
- **χ QCD:** Liu + 15 collaborators
- **Dynamical Chiral:** Edwards + 2 collaborators
- **Lattice SUSY:** Catterall
- **Hazenfratz:** Horvath

Allocations and usage by project

10^6 cpu hrs



Capacity and Usage Comments

- July – November: represents 10 TF-months
- This allocation Year: 34.5 TF-months
- A 30% allocation usage to date is “on track”
- MILClat project is now starting on kaon. It will consume 24,000 core-hrs per day of running and use 42% of kaon’s nodes.
- Heavlight project will soon start 3-pt runs on the fine lattice. This will require bulk of the remaining allocation for this year.

Scientific Progress Reports at LATTICE 2006

- D and B meson decay constants at $a=0.09$ fm and $a=0.15$ fm lattice spacings.
- $B \rightarrow \pi$ $l\nu$ form factors at $a=0.12$ fm lattice spacing.
- $D \rightarrow B^*$ $l\nu$ form factor at $a=0.12$ fm lattice spacing.
- Charmonium and Bottomonium spectra at $a=0.09$, 0.12 , 0.15 and 0.18 fm.
- B-Bbar mixing matrix elements at $a=0.12$ fm.
- HQET parameters Λ_{bar} , λ_1 and charm quark mass.

LQCD Software Frameworks

Project	MILC QMP QDP QIO	Chroma QMP QDP++ QIO	Fermiqcd	Canopy
Heavylight	X Fnal-io		X Fnal-io	X Fnal-io
MILClat	X			
mixBK		X		
NPQCD		X		

Software Frameworks

- Separate library builds for every combination of MPI (infiniband vs myrinet) and compiler (gcc or intel vs portland) users require.
- Separate pion (32-bit ABI) and kaon (64-bit ABI) builds.
- Configure/make for each build, but no workflow to do all library builds on a cluster.
- Minimal or no automatic testing of builds: e.g. “make check”. Differences in MPI implementations and cluster user environments complicate parallel testing.
- SciDAC libraries QMP(MPI), QLA, QDP, QDP++, QIO are installed and in production use on both pion and kaon clusters.
- Chroma also installed and in production use on both clusters.

Production Issues

Primary concerns are cluster (node/network) reliability and I/O related issues ...

- Impact of file corruption caused by bad pion hard disks...
- Role of dcache...
- I/O requirements for analysis of $48^3 \times 128$ lattices...

File corruption

- Caused by defective RAM cache memory in hard drive electronics; Not caught by drive CRC checks!
- Affects only i/o done from newer pion nodes.
- First noticed in data from $40^3 \times 96$ decay constant run. (24 TB of intermediate files)
- Projects involving external users minimally affected? Production ramped up after problem identified / mitigated. Just re-ran after occasional checksum error?
- Decay constant workflow involves up to five separate drives per configuration. Tracking disk provenance means awk / grep scripts to correlate application logs and PBS job logs.
- Fnal-io always checksums data. Used to require user at application level to test checksums. Tests now at library level!
- Checksums persist in .info files separate from data; tests only possible if users propagate .info files.

File corruption assessment

- Final data products potentially tainted by corrupt i/o were identified: data provenance and scan for outliers. Small overlap between outliers and provenance tracking.
- Suspect files from coarse and smaller lattices regenerated. New and old suspect files compared. Typically only a handful of errors affecting 1st or 2nd significant digits. Many errors of smaller magnitudes.
- Subset of affected $40^3 \times 96$ data regenerated. So far, data errors do not appear to shift the values of decay constants.
- Test runs on kaon now being used to rerun some remaining suspect fine lattice production from pion. Also tests application codes on kaon.

Role of Dcache

- “volatile” dcache is used as a distributed cluster file system.
- Production on both clusters now critically depends upon volatile dcache. Potential single point of failure.
- Vexing issue: offline disk cache leads to a stalled dcpp. Batch jobs die when max node-hrs limit reached. Eats resources!
- Concern: large data sets will soon need to be segmented into (many) multiple files written in parallel. How well will dcache perform?
- Rely on public dcache for tape backed storage. Downtime also impacts production on clusters.
- Grid tools: MILClat project and ILDG will need SRM access to both public and volatile dcache. Access rights via lqcd and ildg VOs.

I/O Requirements and “big” lattices

- I/O can no longer be streamed through a single node. For a 48^3 x 128 lattice upwards of 20-30% of runtime can be spent doing single stream i/o!
- sciDAC QIO library supports parallel i/o segmenting data in separate files. Desirable to transition applications to QIO.
- Applications now use /scratch for i/o. dccp used to copy files in/out of dcache. Redundant copies!
- Desirable for applications to do i/o directly from dcache. Need to overload i/o calls in QIO with libdcap replacements.
- QIO modifications done but not folded into QIO release.
- QIO needs to be validated with most recent libdcap release. Need both 32- and 64-bit ABI versions.